Table of Contents

[Github 1](#_Toc406080626)

[Working in local code base 1](#_Toc406080627)

[Building and Running Tests 2](#_Toc406080628)

The SDK code is now located at <https://github.com/hdinsight/azure-sdk-for-net>. This is our public Github repository and is read only to everyone.

# Github

1. Create GitHub account and follow [these instruction](https://help.github.com/articles/set-up-git/) to install Git on your local machine.
2. Create a fork off the HDInsight SDK repo (<https://github.com/hdinsight/azure-sdk-for-net>)

Click the fork button in the top right. Now that you have your own fork of the repository, it is time to pull the content to your local machine. [GitHub Buttons](https://github.com/Azure/adx-documentation-pr/blob/master/Images/github_buttons.png)

1. First, open GitBash (a shortcut would have been installed on your desktop in Step 1 above). Make sure you don't run the program as an administrator.
2. In GitBash type the following into the command line:

**git clone**[**git@github.com**](mailto:git@github.com)**:your-github-username/repository-name.git**

This clones the content of your Fork on GitHub to your local machine.

1. Enter the following in GitBash to move into the source repository.

**cd repository-name**

1. This command configures the HDInsight org fork of the sources as the one to track.

**git remote add upstream** [**git@github.com**](mailto:git@github.com)**: hdinsight /repository-name.git**

1. Get the latest sources from the Azure fork. Probably no change if you just forked your own version.

**git fetch upstream**

This links your local git repository to the source repository on GitHub.

1. Checkout the dev branch

**git checkout –b dev**

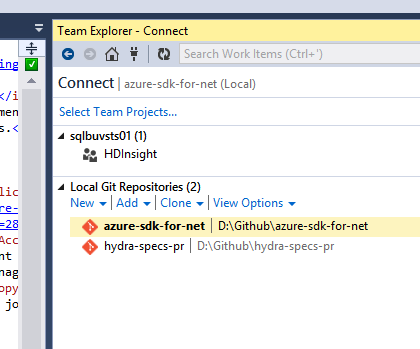
1. Your local repository is now ready to go.
2. Note: HDInsight code location in the common Azure SDK code base:
   1. HDInsight Src code added under:

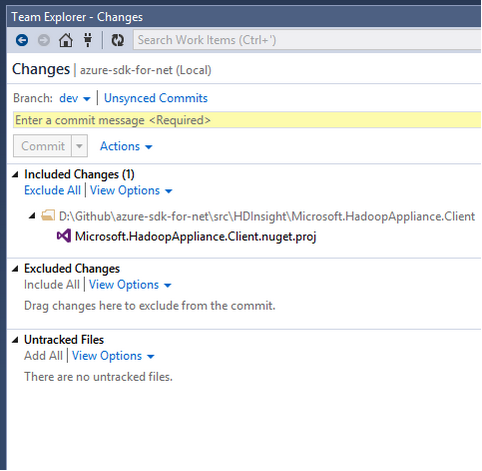
Location: \azure-sdk-for-net\src\HDInsight

* 1. HDInsight solution file that will have all related projects. Devs can open this in Visual Studio and work off of it.

Location: \azure-sdk-for-net\HDInsightManagementLibraries.sln

# Working in local code base

1. Visual Studio 2013 Ultimate GitHub integration will let you setup your local GitHub repo in VS as below. Choose Team Explorer Connect to set the VS to point to your local Git Repo as below.  
   
2. Open the HDInsight solution file from your local repo  
   Ex: D:\Github\azure-sdk-for-net\HDInsightManagementLibraries.sln
3. Tracking changes to your local repo in Visual Studio.
   1. You can use VS to do the following:
      1. Add new files
      2. Check diffs
      3. Commit locally to your repo. This will only check this into your local repo on your machine but does not sync with GitHub repo on the server yet.
   2. Note: Currently VS does not support syncing to remote repo when a remote upstream repo has been configured. You can install GitHub for Windows to push your local changes to the server. Explained in next section (See Section 9).



# Building and Running Tests

Building HDInsight src in Github local repo on your box:

*d:\Github\azure-sdk-for-net>msbuild HDInsightLibraries.msbuild /t:DeveloperBuild;BuildPackages*

Run unit tests on your local Github repo on your box

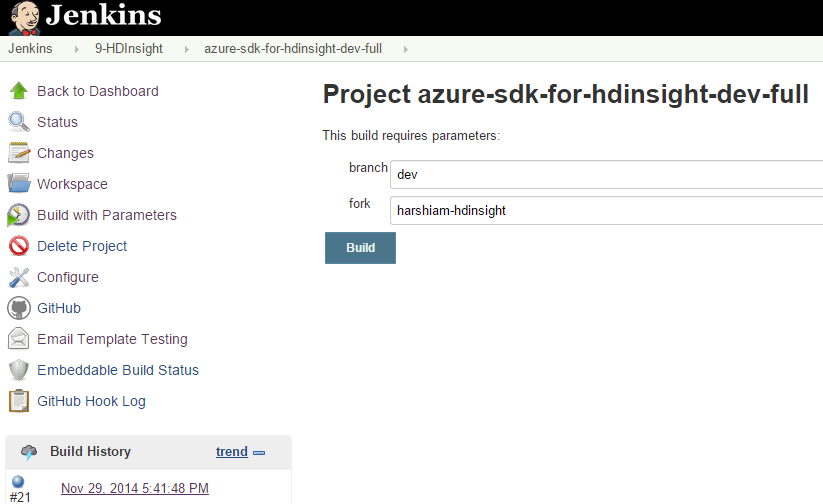
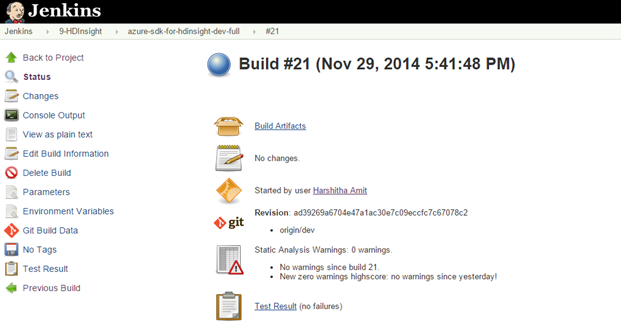
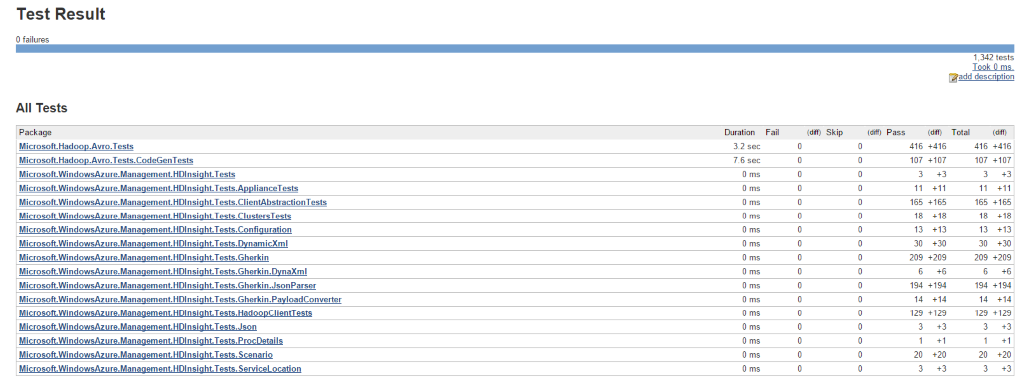
*d:\Github\azure-sdk-for-net>msbuild HDInsightLibraries.msbuild /t:RunTests*

# Jenkins - Kicking off a new build and checking the state

Continuous integration has been setup for HDInsight code base in Jenkins. Once your code changes are checked into your GitHub fork/ branch, you can kick off a Jenkins job to build, publish and run unit tests as follows:

Jenkins Site:  <http://azuresdkci.cloudapp.net/>

Choose HDInsight tab for build jobs : <http://azuresdkci.cloudapp.net/view/9-HDInsight/>

1. Go to the http://azuresdkci.cloudapp.net/view/9-HDInsight/job/azure-sdk-for-hdinsight-dev-full/
2. Choose **Build with Parameters** and provide the parameters for branch and your fork and click Build   
   
3. Check status of the build:
   1. You can check the status of the build in the Jenkins website. Below is a passing build where the build ran with no errors and packages were published and will be available when you click Build Artifacts and also note that unit tests passed with no failures.  
        
        
      
   2. For a failing build, you can look at Console Output to get the build error logs.
   3. Email has been configured. So you will also get an automated email from [azuresdkcheckins@gmail.com](mailto:azuresdkcheckins@gmail.com) with the build log attached.   
      Ex: email for a failing build from Jenkins

Title: azure-sdk-for-hdinsight-dev-full - Build # 20 - Still Failing!

Body: azure-sdk-for-hdinsight-dev-full - Build # 20 - Still Failing:

Check console output at [http://azuresdkci.cloudapp.net:80/job/azure-sdk-for-hdinsight-dev-full/20/](http://azuresdkci.cloudapp.net/job/azure-sdk-for-hdinsight-dev-full/20/) to view the results.

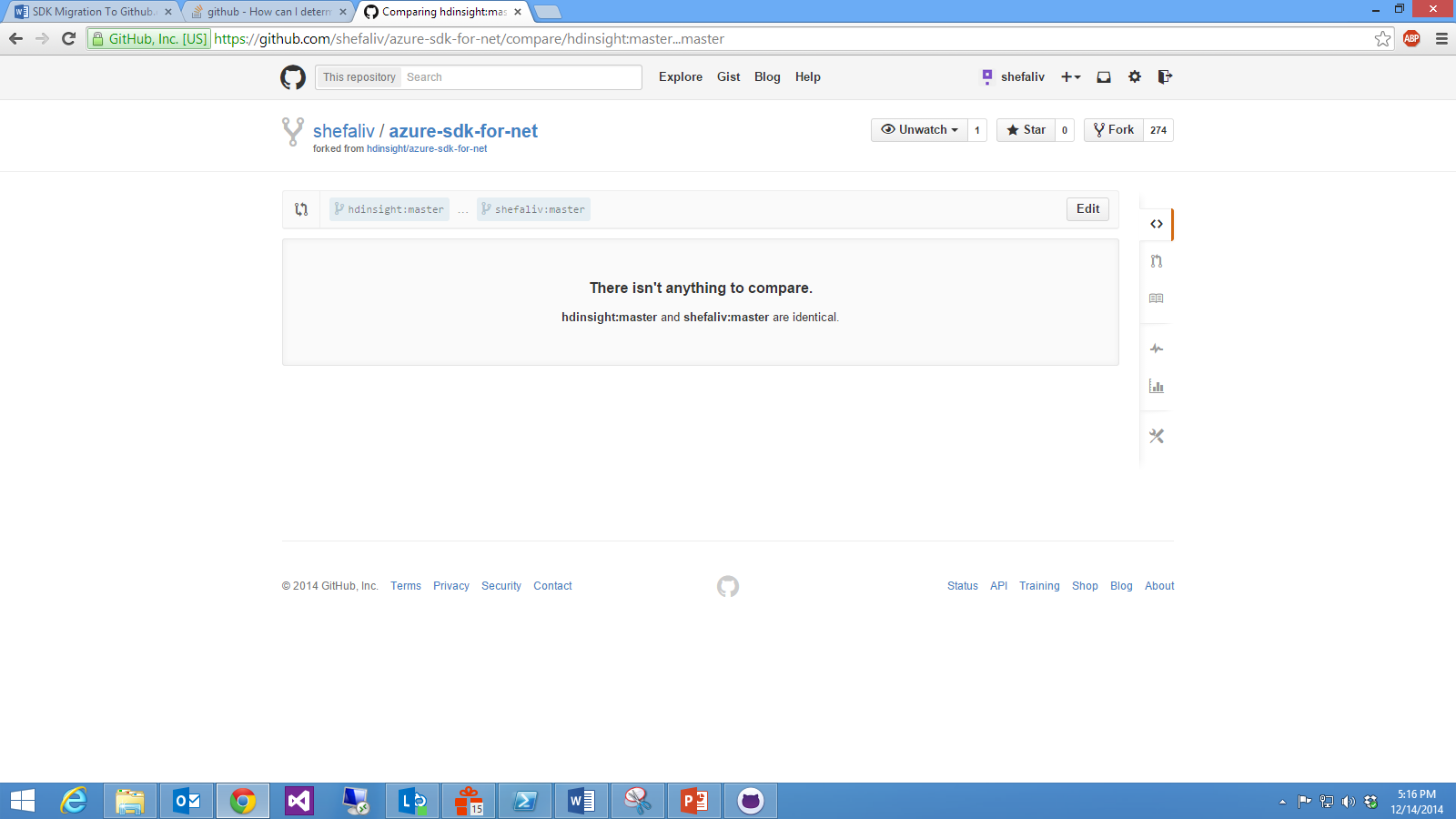
# Code Review Process

Use CodeFlow to send your CR to the right reviewer aliases.

1. Open CodeFlow
2. Choose GitHub as the project
3. Incremental changes can be reviewed by your peers as you work on your Dev branch in your fork. You can publish CRs for review at multiple stages:
   1. Sending CR for locally checked in changes: (origin/dev) **Preferred**
   2. You can send review for all your changes in the Dev branch that is not in the master branch. This is after committing locally and pushing all your changes to your repo.
4. Click Prepare Review
5. Add Tile, Description and Required reviewers (Mahesh, Shefali and Harshitha)

# Submitting a Pull Request

1. Ensure all changes are committed to your fork dev branch.
2. Ensure all changes are pushed to your fork dev branch.
3. Create a new pull request.



1. The branches being compared should be the dev branch on HDInsight and the branch on your fork that you made the changes on. To modify the branches being compared, click the Edit button on the right. This will let you select the forks and branches you want.  
     
   In this case, the changes were on the local dev branch.